Boltzmann Equation for WIMP Dark Matter: Approximate Analytical Solution and Numerical Solution Using Python

Jacinto Paulo Neto

1 Radiation-dominated freeze-out

The evolution of the DM number density n_χ is described by the Boltzmann equation,

$$\frac{dY_\chi(x)}{dx} = -\frac{s(x)\langle\sigma v\rangle}{x\,H(x)}\left[Y_\chi^2(x) - Y_\chi^{\mathrm{eq}\,2}(x)\right] \tag{1}$$

where $x = m_\chi/T$ is a "time" variable that helps us to simplify the integration and physical interpretations and the equilibrium comoving number density

$$Y_\chi^{\mathrm{eq}}(x) = \frac{45}{4\pi^4}\frac{g_\chi}{g_{\star s}}x^2 K_2(x), \tag{2}$$

g_χ corresponds to the DM degrees of freedom and $K_2(x)$ is the modified Bessel function. In this parametrization, the entropy density and the Hubble parameter are

$$s(x) = \frac{2\pi^2}{45}g_{\star s}(x)m_\chi^3 x^{-3}, \tag{3}$$

$$H(x) = \sqrt{\frac{\pi^2 g_\star(x)}{90}}\frac{m_\chi^2}{M_{\mathrm{P}}}x^{-2}, \tag{4}$$

where $M_{\mathrm{P}} = \sqrt{1/(8\pi G)}$ is the reduced Planck mass. Using these equations we can manipulate the right-handed term,

$$\frac{s(x)}{xH(x)} = \frac{2\pi\, g_{\star,s}(x)}{45}\sqrt{\frac{90}{g_\star(x)}}M_{\mathrm{Pl}}\frac{m_\chi}{x^2}. \tag{5}$$

Hence, by rewriting the Boltzmann equation we obtain

$$\frac{dY_\chi}{dx} = -\frac{\lambda(x)}{x^2}\left[Y_\chi^2(x) - Y_\chi^{\mathrm{eq}\,2}(x)\right], \tag{6}$$

where

$$\lambda(x) = \frac{2\pi\, g_{\star,s}(x)}{45}\sqrt{\frac{90}{g_\star(x)}}M_{\mathrm{Pl}}\, m_\chi\,\langle\sigma v\rangle \tag{7}$$

1.1 Approximate DM Yield

Step I Since $n_\chi^{\mathrm{eq}}(x) \propto \mathrm{e}^{-x}$, it decreases much faster than $n_\chi(x)$. Therefore, we can safely approximate the Boltzmann equation to

$$\frac{dY_\chi(x)}{dx} \approx -\frac{\lambda(x)}{x^2}Y_\chi^2(x). \tag{8}$$

Step II We can also assume that at the freeze-out time, $m_\chi \gtrsim 10$ GeV, hence $g_\star \approx g_{\star,s} \approx 106.8$. If we assume s-wave thermal cross-section,

$$\langle \sigma v \rangle \simeq a + \frac{3}{2} b x^{-1} + \mathcal{O}(x^{-2}), \quad v^2 = \frac{3}{2} x^{-1}, \tag{9}$$

only the zeroth-order term a contributes to $\lambda(x)$. Therefore, $\lambda(x) \approx \bar{\lambda}$ does not depend on x. Hence, we rewrite again

$$\frac{dY_\chi(x)}{dx} \approx -\frac{\bar{\lambda}}{x^2} Y_\chi^2(x), \quad \bar{\lambda} = \frac{2\pi}{15} \sqrt{10 g_\star} M_{\rm Pl}\, m_\chi \, \langle \sigma v \rangle. \tag{10}$$

Step III Here, we change variables $\bar{Y}_\chi \equiv 1/Y_\chi$. This results in

$$\frac{dY_\chi}{dx} = \frac{d}{dx}\left(\frac{1}{\bar{Y}_\chi}\right) = -\frac{1}{\bar{Y}_\chi^{-2}} \frac{d\bar{Y}_\chi}{dx} \tag{11}$$

$$\approx -\frac{\bar{\lambda}}{x^2} Y_\chi^2(x) \tag{12}$$

$$\approx -\frac{\bar{\lambda}}{x^2} \frac{1}{\bar{Y}_\chi^{-2}}. \tag{13}$$

On the right-handed side, we equate the first and the last lines to get

$$\frac{d\bar{Y}_\chi}{dx} = \frac{\bar{\lambda}}{x^2}. \tag{14}$$

Now we have to place boundary conditions on this equation somehow we can match the solution in the asymptotic future to a good approximation at $x \sim 1$. Solving this equation, we have

$$\int_{\bar{Y}_{\chi,f}}^{\bar{Y}_{\chi,\infty}} d\bar{Y}_\chi = \bar{\lambda} \int_{x_f}^{\infty} \frac{1}{x^2}, \tag{15}$$

where the subscript f stands for *freeze-out* time and ∞ is the today asymptotic time. Hence, using $Y_\chi = 1/\bar{Y}_\chi$,

$$\frac{1}{Y_{\chi,\infty}} - \frac{1}{Y_{\chi}, f} = \frac{\bar{\lambda}}{x_f}. \tag{16}$$

Typically, we have $Y_{\chi,f} \gg Y_{\chi,\infty}$. Consequently,

$$Y_{\chi,\infty} \approx \frac{\bar{\lambda}}{x_f}. \tag{17}$$

Using the expression for $\bar{\lambda}$, we got the usual formula for the approximate solution of the Boltzmann equation for freeze-out taking place during a radiation-dominated era,

$$Y_{\chi,\infty} \approx \frac{15}{2\pi\sqrt{10 g_\star}} \frac{x_{\rm fo}}{m_\chi M_{\rm Pl} \langle \sigma v \rangle}. \tag{18}$$

1.2 Freeze-out temperature

The freeze-out temperature is obtained via

$$\left.\frac{n_{\mathrm{eq}}\langle\sigma v\rangle}{H}\right|_{x=x_f} = 1, \tag{19}$$

where

$$\begin{aligned} n_{\mathrm{eq}} &\approx \frac{g_\chi}{(2\pi)^{3/2}}\,(m_\chi T)^{3/2} e^{-m_\chi/T} \\ &\approx \frac{g_\chi}{(2\pi)^{3/2}}\frac{m_\chi^3}{x^{3/2}} e^{-x} \end{aligned} \tag{20}$$

Using $H(x)$, s-wave annihilation cross-section, and $x \to x_f$ we obtain

$$e^{-x_f} \approx \sqrt{\frac{8\pi^5 g_\star}{90}}\frac{1}{g_\chi\langle\sigma v\rangle m_\chi M_{\mathrm{P}} x_f^{1/2}}. \tag{21}$$

Using

$$\frac{8}{90} = \frac{4}{45} = \frac{2^2}{3^2\times 5}, \tag{22}$$

Finally, we get

$$x_f \approx -\ln\left[\frac{2}{3}\sqrt{\frac{\pi^5 g_\star}{5}}\frac{1}{g_\chi\langle\sigma v\rangle m_\chi M_{\mathrm{P}} x_f^{1/2}}\right] \tag{23}$$

$$\approx \ln\left[\frac{3}{2}\sqrt{\frac{5}{\pi^5 g_\star}} g_\chi\langle\sigma v\rangle m_\chi M_{\mathrm{P}} x_f^{1/2}\right]. \tag{24}$$

2 Numerical solution – Python

```
import numpy as np
from scipy.integrate import solve_ivp
from scipy.special import kn
import matplotlib.pyplot as plt
from matplotlib import rc
rc('font', **{'family': 'serif', 'serif': ['Computer Modern']})
rc('text', usetex=True)
rc('text.latex', preamble=r'\usepackage{amsmath}')

# Functions
def s(x):
    return (2 * np.pi**2 / 45) * gstar * mchi**3 * x**-3

def H(x):
    return np.sqrt(np.pi**2 * gstar / 90) * mchi**2 / (Mp * x**2)

def Yeq(x):
    return (45 / (4 * np.pi**4)) * (gchi /gstar) * x**2 * kn(2, x)

# Boltzmann equation
def boltzmann_eq(x, Y, sigmav):
```

```
    Yeqx = Yeq(x)
    return -s(x) * sigmav / (x * H(x)) * (Y**2 - Yeqx**2)

Markdown: $\Omega_{\textrm{DM}}h^2 = 2.74385 \times 10^8 \frac{m_{\textrm{DM}}}{\
    textrm{GeV}}  Y_{\textrm{DM}}$

def YDMPlanck(mchi): # DM in GeV
    return (0.12/2.74385)*(1e-8/mchi)

YDMPlanck(100)

# Constants
mchi = 100.0  # GeV
Mp = 2.435e18  # GeV
gchi = 4;  gstar = 106.8

#  Annihilation cross-sections
sigmav_values = [1.8e-10, 1.8e-9, 1.8e-8] # GeV^{-2}

# Initial conditions and integration range
xinit = 1e-2; xend = 1e4; Y0 = Yeq(xinit)

# Solving the differential equation for each cross-section
xvalues = np.logspace(np.log10(xinit), np.log10(xend), 1000)
solutions = {}

for sigmav in sigmav_values:
    sol = solve_ivp(boltzmann_eq, [xinit, xend], [Y0], args=(sigmav,),
        dense_output=True, method='BDF', atol=1e-12, rtol=1e-12)
    Yvalues = sol.sol(xvalues)[0]
    solutions[sigmav] = Yvalues

# Plotting the solution
fig = plt.figure()
fig, ax = plt.subplots(figsize=(7,5), tight_layout=True);

for sigmav in sigmav_values:
    if(sigmav == 1.8e-10):
        plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle
            =10^{-10} \, {\rm GeV}^{-2}$')
    if(sigmav == 1.8e-9):
        plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle
            =10^{-9} \, {\rm GeV}^{-2}$')
    if(sigmav == 1.8e-8):
        plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle
            =10^{-8} \, {\rm GeV}^{-2}$')

plt.plot(xvalues, Yeq(xvalues), label=r'$Y_\chi^{\rm eq}(x)$', linestyle='--',
    color='black')
#plt.axhline(y=YDMPlanck(100), color='r', linestyle='-')
fig.text(0.2, 0.32, r"$\Omega_{\textrm{DM}} h^2 \approx 0.12$", fontsize=12, color
    ='gray')
plt.axhspan(YDMPlanck(100)+1.2e-12, YDMPlanck(100)-1.2e-12, facecolor ='gray',
    alpha = 0.3)
plt.xscale('log')
plt.yscale('log')
plt.xlabel(r'$x = m_\chi / T$', fontsize=14)
plt.ylabel(r'$Y_\chi(x)$',fontsize=14)
plt.xlim(1,1e4)
```

```
plt.ylim(1e-14,1e-1)
plt.legend()
plt.savefig('WIMPDM-BEQ.pdf')
plt.show()
```

Listing 1: Solution of the Boltzmann Equation for WIMP Dark Matter

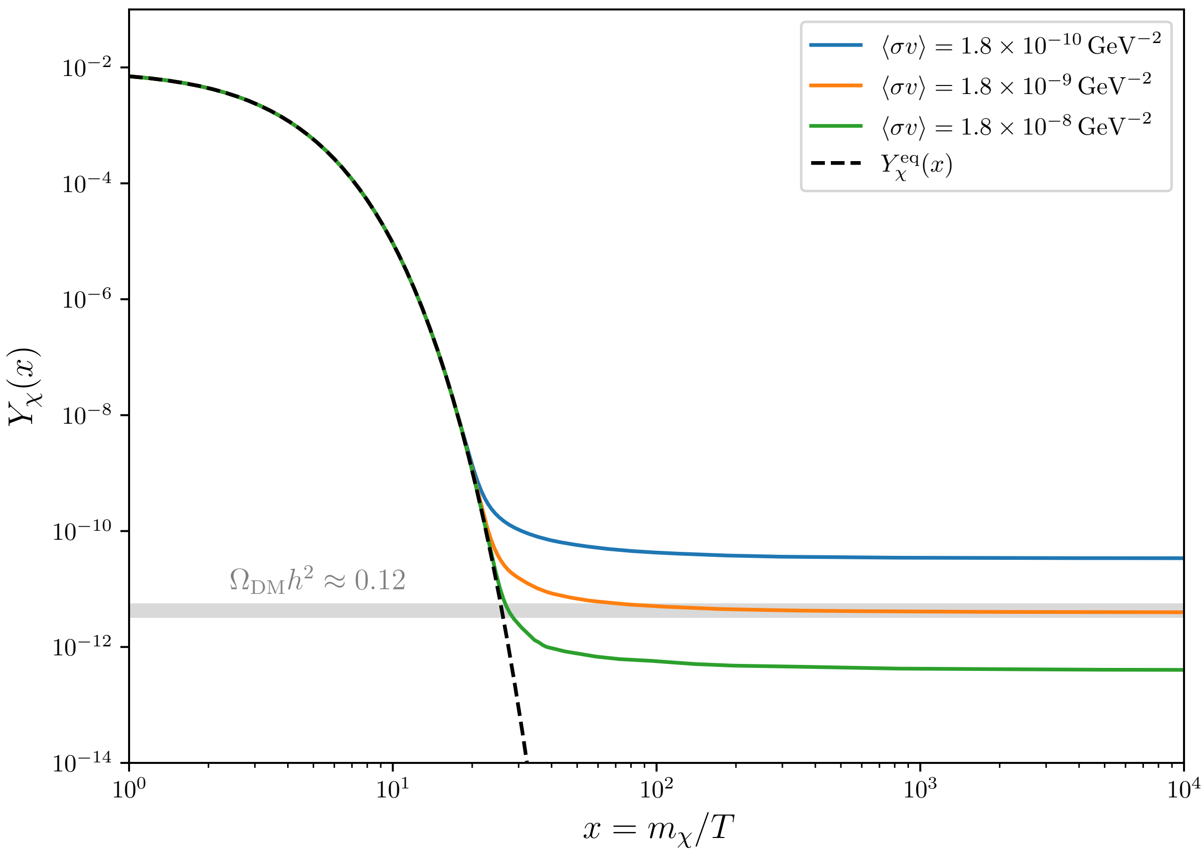

Figure 1: Numerical solution of the Boltzmann equation for WIMP Dark Matter.